

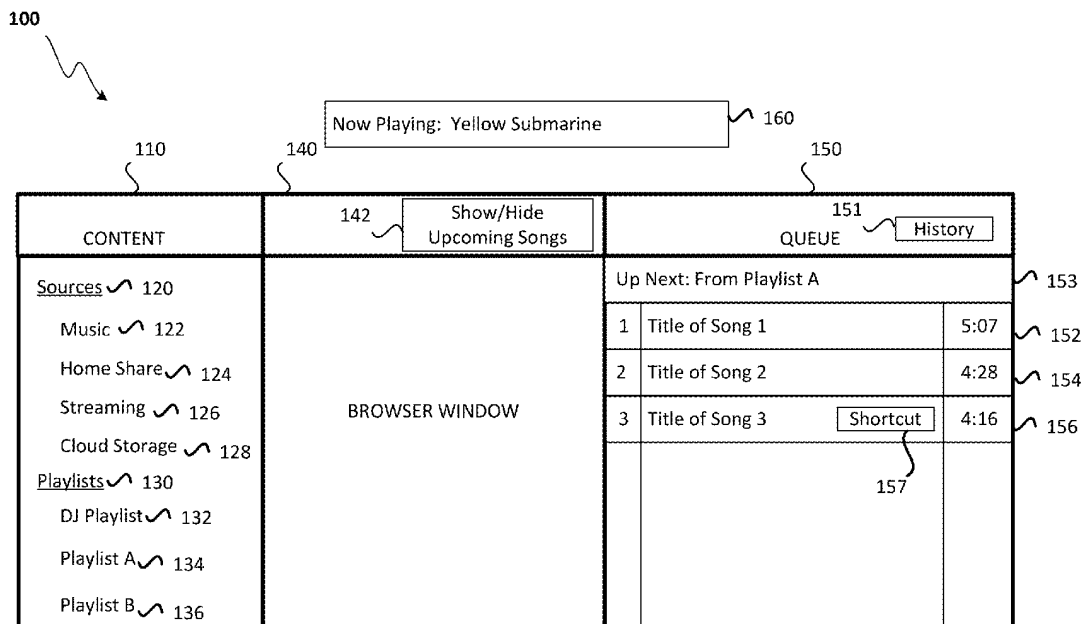
# EXHIBIT 12

## (Excerpted)

US 20140075308A1

(19) **United States**(12) **Patent Application Publication**  
**Sanders et al.**(10) **Pub. No.: US 2014/0075308 A1**(43) **Pub. Date: Mar. 13, 2014**(54) **INTELLIGENT MEDIA QUEUE****Publication Classification**(75) Inventors: **Christopher John Sanders**, San Jose, CA (US); **Anne Jones**, Emerald Hills, CA (US); **Jeffrey L. Robbin**, Los Altos, CA (US); **William Martin Bachman**, San Jose, CA (US); **Timothy B. Martin**, Sunnyvale, CA (US)(51) **Int. Cl.**  
**G06F 3/01** (2006.01)(52) **U.S. Cl.**  
USPC ..... 715/716(57) **ABSTRACT**

Systems, methods, and non-transitory computer-readable storage media for intelligently managing a playlist of digital media provide an intelligent dynamic queue that is configured to manage the playback of digital media. The queue can transition between passive playback mode, active playback mode, and mixed playback mode. The queue can handle the playback of the songs in the queue according to the playback mode and/or a queue status field that is associated with each song in the queue.

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)(21) Appl. No.: **13/609,169**(22) Filed: **Sep. 10, 2012**

US 2014/0075308 A1

Mar. 13, 2014

2

[0020] FIGS. 10a and 10b illustrate an example of a queue with a history list. The exemplary queue can be presented on a graphical user interface;

[0021] FIG. 11 illustrates an exemplary method for adding a song to a location in the queue;

[0022] FIG. 12 illustrates an exemplary method for adding a song as a next song in the queue;

[0023] FIG. 13 illustrates an exemplary user interface of a media playback application;

[0024] FIG. 14 illustrates an exemplary mini player user interface of a media playback application;

[0025] FIG. 15 illustrates an exemplary system; and

[0026] FIG. 16 illustrates an exemplary cloud system.

#### DETAILED DESCRIPTION

[0027] Various embodiments of the disclosure are discussed in detail below. While specific implementations are discussed, it should be understood that this is done for illustration purposes only. A person skilled in the relevant art will recognize that other components and configurations may be used without parting from the spirit and scope of the disclosure.

[0028] The present disclosure addresses the need in the art for systems, techniques, and methods for improved playback management of digital media. The digital media can include digital songs, movies, videos, podcasts, music videos, images, photos, and other digital media content. While the disclosure focuses on playback management for digital songs, it is to be understood by a person of ordinary skill in the relevant art that these teachings also apply to playback management for other forms of digital media such as movies, images, videos, photos, podcasts, etc.

[0029] The present disclosure focuses on an intelligent queue that is used by a media playback application to manage the playback of digital media items. An intelligent queue is different than a traditional playlist. An intelligent queue can contain media items from multiple media sources, some of the sources being remotely located. If one of the remote sources were to go offline, the integrity of the queue is not affected since the intelligent queue is configured to skip songs that are offline. In contrast, a traditional playlist generally contains only items that are locally stored on a playback device. Playlists can sometimes play back media from multiple sources however, when one of the sources go offline, the playlist's integrity is lost. Moreover, an intelligent queue can be dynamically modified by adding other media items into the queue in real time whereas modifications to a playlist are persistent in that they are stored in memory. Furthermore, the intelligent queue is capable of interpreting the user's intentions based on where the user adds digital media into the queue and as a result, affect how other media items in the intelligent queue are handled.

[0030] In some examples, different types of digital media can co-exist in a single queue. For example, a queue can include a song and a video that are presented in a specified playback order. A user can submit requests to a media playback application to add songs (or other types of digital media) from a single source or from various sources to the queue. Each song can be linked or otherwise associated with an entry in the queue. Besides adding an individual digital media item, entire playlists, albums, and other collections of music can be selected for addition to the queue. When the collection of music is added, the digital media within the collection are added to the queue. Playback of music by the media playback

application can be based on the queue and more specifically, the order that the songs appear in the queue. Since the user can in real time add digital media to the queue, the queue is dynamic and changes constantly based on the user's desires. In some embodiments, the queue is also persistent in that un-played tracks that are in the queue when the media playback application closes will remain queued for playback the next time the media playback application opens.

[0031] Through the intelligent queue, the media playback application is capable of seamlessly changing in between two different modes of operation. In a first mode, the media playback application operates in a normal playback mode. In the normal playback mode, the media playback application begins by playing the first song in the playlist and continues to sequentially play through the tracks in the playlist until the last song is played. In a temporary playback mode, songs can be dynamically entered and played back in a specified order. When the list is done, playback is completed and the list is cleared. Traditionally these different modes of operation have been mutually exclusive however through the use of the intelligent queue, the media playback application is able to seamlessly switch between the two modes according to actions items and triggering events initiated by the user. For example, dragging a song into the queue can result in the media playback application transitioning from a normal playback mode to the temporary playback mode. Once the song has completed, the media playback application can transition back to the normal playback mode. Therefore, the media playback application is capable of operating in a normal playback mode that does not require user interaction and then subsequently change to a temporary playback mode when a request is received to listen to a song. Listening to the song does not abandon the normal playback mode. Instead, the media playback application returns to the normal playback mode when the song has completed.

[0032] The media playback application can associate queue status fields to a song when the song is added to the queue. More specifically, each entry in the queue can store information related to a song. For example, the information can include a link to the song and a queue status field of the song. Depending on the queue status field associated with the song, the song can be handled differently by the media playback application. In one embodiment, each entry can include a queue status field configured to identify whether a song was manually entered or automatically entered. Manually entered songs can remain in the queue when the source of music changes. As another example, songs that were automatically entered into the queue can be purged from the queue when the source of music changes. The media playback application interprets manually entered songs as songs that the user has explicitly selected to listen to, either now or in the future. In contrast, songs that are automatically entered (e.g., queuing songs from an internet radio station such as Pandora® or queuing songs from a playlist) generally belong to a collection of songs that share a commonality, the commonality being that the songs all belong to a collection that was selected by the user to be added to the queue. Since this commonality is the reason that the songs were added to the queue (and not because the user has manually selected each song to be added into the queue), these songs are volatile and in some instances can be deleted from the queue when the source changes or when the media playback application closes. Examples of songs that are automatically entered into the queue include the situation where a user selects a playlist

US 2014/0075308 A1

Mar. 13, 2014

3

to be added to the queue or where the user requests the media playback application for ten songs that are similar to a selected seed song. The recommended songs are added to the queue. In some examples, the media playback application can include rules for editing the behavior of the media playback application, such as setting the circumstances when the type (manually entered or automatically entered) should be changed in an entry. Additional details about automatically entered and manually entered songs are discussed below.

#### Media Playback Application

[0033] FIG. 1 illustrates an exemplary media playback application. The media playback application can be an application running on a client device that is capable of playing back digital media. Exemplary media playback applications include the iTunes® media player. Media playback application 100 includes content bar 110, browser window 140, queue bar 150, and currently playing bar 160. Content bar 110 is configured to display music that is available to the user. Using browsing window 140, a user can browse the music from a source and select one or more songs that the user would like to listen to. Selected songs can appear in queue bar 150 for scheduled playback. Queue bar 150 can be hidden or displayed using show/hide upcoming songs button 142. In some examples, queue bar 150 can be a pop up screen rather than a side bar, where the pop up screen can be activated or deactivated through a display option such as show/hide upcoming songs button 142.

[0034] In one embodiment, content bar 110 can be organized into sources 120 and playlists 130. Sources 120 display a list of sources of music that are available to the user. Here, the sources include local music library 122 (e.g., music library that the user has stored locally on his device), home share 124 (e.g., music from the music libraries of other users on the same network), streaming 126 (e.g., music that the user streams from remote sources, such as internet radio stations), and cloud storage 128 (e.g., music stored remotely that the user has rights to play or is otherwise associated with the user). Available music that is not stored on the user's local device can be accessed through Wifi, Bluetooth, Intranet, Internet, and other connection options. In other embodiments, other types of media can exist in the sources, such as movies and TV shows. However for simplicity, only audio tracks are discussed. A person of ordinary skill in the art would be able to expand the teachings here to other types of media, such as movies, images, podcasts, videos, photos, etc.

[0035] Playlists 130 shows a list of playlists that are available to the user. The playlists can be generated by the user or automatically generated by a third-party service, a remote server, or the media playback application. Here, playlists 130 include DJ playlist 132, playlist A 134 and playlist B 136. Playlist A 134 and playlist B 136 may include a collection of songs that are available to the user from sources 120. DJ playlist 132 can include a collection of songs that was automatically generated by the media playback application or a remote server. In one embodiment, the media playback application automatically creates the collection of music based on parameters provided by the user. For example, top rated songs can be added and prioritized by placing them in a higher queue position than other songs. In some examples, DJ playlist 132 can be dynamically generated whenever the playlist is selected. This can allow DJ playlist 132 to take advantage of up to date metadata when automatically selecting the songs to

play. In other examples, DJ playlist 132 can be generated or regenerated based on user request.

[0036] Queue 150 displays an ordered list of songs for playback. The songs in the list can be from multiple sources or multiple playlists. The order that the songs appear in queue 150 is the playback sequence of the queue. When a currently playing song is finished or about to be finished playing on a media player device, the next song in the queue is retrieved for playback. For media stored remotely (i.e., not stored on the user's device), the media playback application can pre-fetch a predetermined number of songs in the queue from the remote source and store them locally for future playback. This may improve performance by reducing the opportunity for the music to refresh or lag during playback. The retrieved song is removed from the ordered list and the remaining songs in the list are promoted up one cell. Since songs can be dynamically selected and added into the queue, queue 150 also is dynamic in nature. Thus, a user can add songs into the queue. In other examples, other people using other devices can add songs to the queue. Queue 150 can also be persistent, thus allowing songs that were scheduled for playback to be presented to the user the next time the user starts the media playback application.

[0037] Here, queue 150 includes entries 152, 154, and 156, each identifying a song from a particular media source. The three songs 152, 154, and 156 are located under heading 153 titled "Up Next." The songs are presented in a specified playback order. Thus, the next song to be played is the song identified by entry 152, followed by the song identified by entry 154, followed by the song identified by entry 156. In some examples, heading 153 can also present other information associated with the origin of this grouping of songs. For example if the songs are all part of a playlist, the name of the playlist can also be presented in heading 153. This can be useful to provide additional information about the collection of music that the user is currently listening to. In some examples, each entry 152, 154, 156 can display additional information about the song in an auxiliary view that is displayed when a cursor is placed over the song. For example, queue 150 may by default be configured to display the title of the song and the duration of the song. However when the cursor is placed over the entry, song metadata such as lyrics, artist, genre, rating, etc. can also be displayed in an auxiliary view. In other examples, hidden action items that can be performed on the entry can also be presented when the cursor is placed over the song or the cursor rolls over the song/entry. For example, action items such as skip song, promote song to play next, promote song to play immediately, rate song, create a playlist by using the song as a root song, and others can be presented to the user when the cursor is placed over the song/entry. In other examples, one or more of these action items can be fixed on the entry and presented along with the default information displayed. Here, entry 156 presents shortcut 157 along with the title and duration (i.e., default information). Shortcut 157 can be linked to a predefined or user defined action item associated with the song such as skip song or move song to play next. In some examples, shortcut 157 can appear on a rollover.

[0038] Queue 150 can also include a history icon 151. History icon 151, when selected, presents a list of songs that have been previously played by the media playback application. The list of previously played songs can include songs from different sources and playlists. In one embodiment, a predefined minimum play time can be applied to songs that